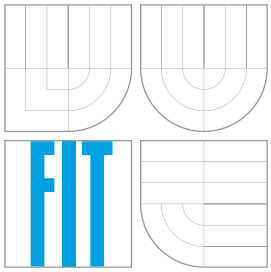
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKY AKTUALIZOVANÝ WEBOVÝ PORTÁL

AUTOMATICALLY UPDATED WEB PORTAL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR STANĚK

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2016

Zadání bakalářské práce/18568/2015/xstane34

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Staněk Petr**
Obor: Informační technologie
Téma: **Automaticky aktualizovaný webový portál**
Automatically Updated Web Portal
Kategorie: Web

Pokyny:

1. Seznamte se s existujícími vědeckými portály a zpracujte přehled vlastností, které charakterizují databáze a portály zaměřené na výzkumné projekty.
2. Navrhněte systém automatické aktualizace portálu, který bude reagovat na nalezení nových záznamů metavyhledáváním pomocí dostupných rozhraní webových vyhledávačů a zpracováním získané informace.
3. Implementujte navržený systém s důrazem na výkonnost systému aktualizací.
4. Připravte demonstrační data, která dovolí zhodnotit uživatelskou stránku prezentační části systému.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Funkční prototyp řešení

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Smrž Pavel, doc. RNDr., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2015
Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato bakalářská práce se věnuje návrhu a implementaci automaticky aktualizovaného webového portálu, který řeší nedostatky portálů plněných obsahem lidmi. Dále předkládá srovnání existujících vědeckých portálů, rozebírá problematiku extrakce, ukládání a vyhledávání informací. Obecné mechanismy jsou demonstrovány na portálu evropských výzkumných projektů, který odstraňuje nedostatky oficiálního informačního portálu pro evropský výzkum a inovace Cordis. Práce bere jako prototyp existující produkt bakalářské práce a jejím cílem je vylepšit kvalitu extrakce a rozšířit tento systém tak, aby zjišťoval případné problémy a upozorňoval na ně administrátora. Toho bylo dosaženo zvýšením robustnosti a rychlosti extraktoru, evidováním všech důležitých událostí spojených s extrakcí a na druhé straně implementací samostatné administrační sekce webového portálu, která administrátora informuje o problémech a nabízí mu prostředky k jejich řešení.

Abstract

This bachelor's thesis is dedicated to the design and implementation of an automatically updated web portal that tries to resolve the shortcomings of the portals filled with other people's content. Furthermore, it presents a comparison of the existing scientific portals, it discusses the problems of extraction, saving and searching for information. General mechanisms are demonstrated on the European research projects portal, which removes the shortcomings of CORDIS, the official information portal for European research and development. The thesis takes the existing product as a prototype and its aim is to improve the quality of the extraction and extend the system to detect any potential problems and notified an administrator of them. This was achieved by increasing the robustness and speed of the extractor, by registering all the important events associated with the extraction and, on the other side, the implementation of the separate administrator section of the web portal, which informs the administrator about problems and offers the problem-solving devices.

Klíčová slova

web, portál, vyhledávání, fasetové vyhledávání, extrakce informací, databáze, administrace, Python, Flask, Elasticsearch

Keywords

web, portal, search, faset search, information extraction, database, administration, Python, Flask, Elasticsearch

Citace

STANĚK, Petr. *Automaticky aktualizovaný webový portál.* Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Smrž Pavel.

Automaticky aktualizovaný webový portál

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph. D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Staněk
17. května 2016

Poděkování

Děkuji doc. RNDr. Pavlu Smržovi, Ph.D. za aktivní odbornou pomoc a podněty při tvorbě této práce.

Obsah

Kapitola 1

Úvod

1.1 Motivace

V dnešní době jsou pro lidi informace dostupné prostřednictvím webových portálů velmi důležité. Většina portálů je však založena na manuálně aktualizované databázi, která je plněna, spravována a aktualizována lidmi. Toto má ale omezení v často nedostatečném rozsahu informací dostupných v rámci jedné databáze, případně nutnosti pravidelně aktualizovat navázané informace, jakmile se objeví nové položky.

Toto téma jsem si vybral, jelikož mám zkušenosti s tvorbou webových stránek a chtěl jsem je uplatnit při implementaci portálu a také se naučit novým technologiím a principům.

1.2 Cíl práce

Cílem této práce je vytvořit webový portál, který bude aktualizován automatizovaně. Aby byly ukázány obecné principy, je v práci implementován portál evropských výzkumných projektů, který řeší nedostatky oficiálního informačního portálu pro evropský výzkum a inovace Cordis. Uvedený portál plní obsahem pracovníci Evropské komise. Mnoho vědeckých zpráv zde chybí a ty, které nezanesou do zmiňovaného systému, se často po skončení projektu a případném zrušení webové stránky projektu stávají nedostupnými.

Moje práce bere jako prototyp bakalářskou práci *Automaticky aktualizovaný webový portál o evropských výzkumných projektech* Lucie Dvořákové [4], jejíž kód byl sice z převážné části přepsán, avšak většina jejích myšlenek zůstala zachována. Cílem práce je oproti původnímu stavu vylepšit kvalitu extrakce a zvýšit počet nalezených dat, rozšířit systém o automatické zjišťování problémů a jejich interpretaci administrátorovi, zdokonalit fulltextové vyhledávání, nabídnout komplexní statistiky o datech uložených v databázi a umožnit správu projektů včetně nutných ručních oprav automaticky přiřazených položek.

1.3 Členění práce

V kapitole 2 s názvem Rozbor řešené problematiky vysvětluji pojem portál, předkládám srovnání existujících vědeckých portálů, rozebírám problematiku extrakce, ukládání a vyhledávání informací. V následující kapitole 3 nazvané Návrh a implementace systému se zaměřuji na analýzu problémů, implementaci nalezení a zpracování projektů, rozhraní a implementaci portálu. V předposlední kapitole 4 Vyhodnocení systému prezentuji úspěšnost a přesnost systému a v poslední 5. části pojmenované Závěr shrnuji tuto práci.

Kapitola 2

Rozbor řešené problematiky

2.1 Webové portály

Webové (nebo též internetové portály) představují vstupní bránu k nepřebernému množství informací, které jsou jinak roztroušeny po Internetu, a tak se stávají těžko dostupnými a vyhledatelnými. Zprostředkovávají jednotný přístup k informacím získaných z různých zdrojů a umožňují tyto informace lépe a rychleji třídit i vyhledávat.

2.1.1 První portály

Prvně se pojem portál začal používat pro internetové vyhledávače, jejichž primární účel spočíval ve fulltextovém indexování těžko přístupného obsahu na Internetu a poskytování fulltextového vyhledávání. Tyto vyhledávače na počátku devadesátých let otevřely jejich uživatelům dveře do neustále rostoucího světa Internetu.

S příchodem obrovských populárních vyhledávačů jako jsou například Yahoo!, Excite a Lycos se výraz portál začal používat pro označení těchto velkých stránek, které poskytovaly, na rozdíl od prvotních vyhledávačů vyhledávajících pomocí klíčových slov v obsahu celého Internetu, organizovanou strukturu kategorií umožňující lépe nalézt relevantní webové stránky [15].

2.1.2 Portály dnes

Dnes se prvotní význam slova portál v souvislosti s moderními strategiemi internetových vyhledávačů vytrácí. Přidáváním nových služeb jako jsou zpravodajství, počasí, kalendář, informace o kurzech, diskuzní skupiny a e-mailové služby se jejich provozovatelé snaží zdržet uživatele co nejdéle v prostředí portálu, a to především z marketingových důvodů [3].

2.1.3 Základní dělení portálů

Portály můžeme podle rozsahu poskytovaných informací rozdělit na dvě základní skupiny, a to na horizontální a vertikální.

Horizontální portály nabízejí přístup k širokému spektru informací z mnoha odvětví a cílí na širokou skupinu uživatelů. Mohou existovat v mnoha jazykových mutacích, a tak poskytovat své služby uživatelům celosvětově. Dobrým příkladem je portál internetového giganta Google [10].

Vertikální portály se naopak zaměřují na konkrétní kategorii informací a slouží pro určitou skupinu uživatelů. Může se jednat například o zpravodajské, obchodní nebo vědecké portály [10].

2.2 Existující vědecké portály

Podle studie Digital Universe organizace IDC z roku 2011 se objem světových dat každé dva roky více než zdvojnásobí, a tedy roste dokonce rychleji, než udává Moorův zákon [8].

Právě obrovské množství online dostupných informací je výzvou pro vznik celé řady portálů, jejichž cílem je vesměs usnadnit nalezení požadovaných informací a oprostit uživatele od zbytečného balastu. Mnoho portálů spojuje myšlenka otevřeného přístupu k informacím, která spočívá ve volném přístupu k obsahu, metadatům a jejich agregaci a další distribuci.

Vzhledem k povaze této bakalářské práce se v následujících odstavcích zaměřím na portály spravující informace z vědecké oblasti.

2.2.1 CORDIS

CORDIS[1] (Community Research and Development Information Service) je informační portál pro evropský výzkum a inovace spuštěný roku 1994. Slouží jako primární portál Evropské komise, který je zdrojem informací o Evropskou unií financovaných výzkumných projektech a jejich výsledcích, a také jako prostředí k nalezení partnerů pro výzkum. Poskytuje v první řadě všechny veřejné informace o projektech (základní informace, vědecké zprávy a souhrnné zprávy) a v druhé řadě novinky, události, magazíny a další. Pro registrované uživatele nabízí odběr novinek a tišených publikací zaměřených na výzkum.

Portál je plněn a spravován úředníky Evropské komise, kteří zodpovídají za zveřejňovaný obsah. Rozhraní vyhledávače nabízí bohaté pokročilé vyhledávání. Toto vyhledávání je rozděleno zvlášť na obecné vyhledávání v celém obsahu webu a na velmi podrobné vyhledávání pouze mezi projekty, a to vše s podporou našeptávání možných hodnot jednotlivých atributů. Pokročilejší uživatelé mohou rychle vyhledat požadované informace s využitím speciální syntaxe dotazovacího jazyka zadávaného do samostatného textového pole. Nalezené výsledky lze filtrovat použitím fasetového vyhledávání (předmět, program, země, typ obsahu a jazyk) a dále pak seřadit na základě data poslední aktualizace nebo titulku. Výsledky je možné poté vyexportovat do strojově zpracovatelných formátů. Informace o konkrétním zvoleném projektu lze navíc stáhnout i ve formátu PDF. Tyto formáty nabízí prostředky pro opětovné použití informací, což koresponduje s podporou otevřeného přístupu a volného opětovného využívání obsahu. Nakonec nutno dodat, že je dlouhodobě znatelná velmi pomalá odezva serveru, především při vyhledávání.

2.2.2 Semantic Scholar

Semantic Scholar[2] je vyhledávač navržený pro inteligentní vyhledávání výzkumných publikací z oblasti počítačových věd, který je v provozu od konce roku 2015. Využívá několika inovativních metod, kterými se snaží dosáhnout přesnějších výsledků vyhledávání, než zvládá konkurence. V současné době je stále v beta verzi a ve vývoji, a proto zatím nabízí poměrně strohé uživatelské rozhraní.

[1] CORDIS: http://cordis.europa.eu

[2] Semantic Scholar: https://www.semanticscholar.org/

Jádrem portálu je *crawler*[3], který hledá a stahuje veřejně dostupné dokumenty, ze kterých extrahuje metadata a ta dále analyzuje sofistikovanými metodami, stejně tak jako tomu je u mnoha dalších portálů podobného charakteru. Semantic Scholar na získaná data navíc aplikuje pokročilé techniky strojového učení a zpracování přirozeného jazyka. Rozlišuje několik druhů citací, analyzuje kontext použité citace, důležitost daného dokumentu mezi ostatními dokumenty, extrahuje obrázky, tabulky a schémata. Pro zobrazení výsledků vyhledávání využívá nového způsobu řazení výsledků na základě průměrného počtu citování daného dokumentu za dobu posledních tří let. Samotné vyhledávání je maximálně jednoduché, podle FAQ je to dokonce záměr. Nejsou podporovány ani již dnes naprosto běžné logické operátory ani jiná forma složitějšího dotazu. Toto omezení je kompenzováno pomocí šesti filtrů. Zajímavé jsou z nich zejména tři, a to filtr nazývaný *Overviews* umožňující volbu mezi články a přehledy, *Data set used* představující zvolenou datovou sadu využitou při strojovém učení a *Key Phrase* umožnující filtrovat výsledky na základě nefrekventovanějších řetězců extrahovaných z dokumentů.

Uživatele na první pohled upoutá nezvykle jednoduchý design a struktura portálu. Nutno podotknout, že se zatím jedná pouze o beta verzi a v budoucnu se může tento koncept trochu změnit.

2.2.3 Google Scholar

Google Scholar[4] je volně dostupný vyhledávač odborných a vědeckých prací napříč všemi vědeckými disciplínami poskytující své služby již od roku 2004. Pyšní se obrovským rozsahem a univerzálností databáze. Zahrnuje časopisecké a sborníkové články, preprinty, technické zprávy, odborné knihy, vědecké kvalifikační práce, patenty atd. V systému se nacházejí jak dokumenty volně dostupné na webu, tak i od akademických nakladatelů, z katalogů odborných knihoven, otevřených i uzavřených fulltextových databází, repozitářů společností a univerzit a další. Zmíněný obsah je získáván automaticky systematickým procházením zdrojů robotem. V případě neveřejných, komerčních dokumentů je přístup k obsahu podmíněn zaplacením poplatku [2].

Rozhraní vyhledávače se podobá dobře známému vyhledávání použitém ve vyhledávači Google. V rozšířeném vyhledávání lze navíc kromě klasických atributů specifikovat jméno autora, místo a datum publikace. Nalezené výsledky lze dále v postranní nabídce třídit podle roku. Jejich řazení vychází z relevance založené na počtu citací a dalších kritérií. V případě potřeby lze výsledky seřadit také podle data. Pokud chce mít uživatel přehled o nových dokumentech, které ho zajímají, může si nastavit k tomuto účelu určené notifikace.

2.2.4 ACM Digital Library

ACM Digital Library[5] nabízí přístup k plným textům časopisů, sborníků, knih, magazínů a zpravodajů americké počítačové společnosti ACM a dále pak hostuje fulltextové publikace některých dalších vybraných vydavatelů. ACM DL na rozdíl od většiny ostatních nedovoluje přístup robotům, kteří stahují články a metadata.

Pokročilé vyhledávání nabízí velké množství atributů, které lze využít pro nalezení požadovaného díla. Nalezené výsledky lze dále filtrovat pomocí několika flitrů a seřadit je podle relevance, data publikace, počtu citací nebo počtu stažení. Po registraci je možné také nastavit zaslání novinek.

[3] robot, který systematicky prohledává web za účelem vydolování informací

[4] Google Scholar: https://scholar.google.cz/

[5] ACM DL: http://dl.acm.org/

2.2.5 IEEE Xplore Digital Library

IEEE Xplore Digital Library[6] zpřístupňuje vědecký a technický obsah publikovaný institutem IEEE (Institute of Electrical and Electronics Engineer) a jeho partnery. Jedná se o články, materiály z konferencí a normy z oblasti elektrotechniky a informatiky.

Portál nabízí velmi komplexní rozšířené vyhledávání. Méně pokročilý uživatel může svůj dotaz formulovat pomocí formulářů, kde uvede požadovaná klíčová slova a fráze, případně atributy citace, čímž svůj požadavek dostatečně upřesní. Pokročilý uživatel ocení průvodce tvorby dotazů, který pomáhá vytvořit libovolně složitý dotaz. Nalezené položky umí navíc dále filtrovat na základě klíčového slova a pomocí fasetového vyhledávání. Výsledky lze seřadit dle relevance, data, počtu citování a také abecedně. Registrovaný uživatel může využít historii vyhledávání, exportovat seznam výsledků nebo konkrétní dokumenty do různých formátů, nastavit upozornění na nové dokumenty pro požadovaný dotaz a mnoho dalšího.

Portál na první pohled působí velmi prakticky. V porovnání s ostatními prezentovanými portály se jeho vyhledávání jeví jako jedno z nejvíce propracovaných.

2.2.6 Elsevier ScienceDirect

Elsevier ScienceDirect[7] je databáze zahrnující vědecké časopisy a monografie, především z oblasti medicíny, přírodních věd, informatiky, ekonomie, psychologie a sociálních věd, částečně produkované vydavatelstvím Elsevier. Přístup k plným textům je obvykle podmíněn zaplacením předplatného.

Samozřejmostí je pokročilé i expertní vyhledávání, které umožňuje sestavovat složitější dotazy pomocí logických operátorů a klíčových slov. Rozhraní vyhledávače dovoluje filtrovat výsledky pomocí několika faset, odstranit výsledky bez veřejně přístupného plného textu, řadit nalezené dokumenty dle relevance nebo data publikování a nastavit upozornění na nové dokumenty odpovídající zadanému dotazu.

2.2.7 SpringerLink

SpringerLink[8] je platforma poskytující přístup k vědeckým dokumentům především z oblasti vědy, techniky a medicíny spuštěná v roce 1996. Zahrnuje časopisy a knihy z produkce vydavatelství Springer-Verlag a některých dalších vydavatelství, referenční příručky a protokoly. Většina obsahu je dostupná pouze pro předplatitele.

Obsah dokumentů je automaticky analyzován pro potřeby poskytování sémantického linkování. Díky této vlastnosti systém sám nabízí další relevantní dokumenty, které by mohly uživatele zajímat. Rozšířené vyhledávání v porovnání s ostatními portály není moc pestré, lze vyhledávat jen pomocí klíčových slov ve spojení s logickými operátory. Je velmi podobné tomu, které známe z vyhledávače Google. Pročistit nalezené výsledky na základě dalších parametrů lze až po jejich zobrazení pomocí jednoduchých faset – typ obsahu, vědní obor a podobor a jazyk. Úspěšnějšímu nalezení požadovaných informací napomáhá řazení dle relevance, případně dle data. O nových dokumentech v databázi může být uživatel informován pomocí notifikací.

[6] IEEE Xplore DL: http://ieeexplore.ieee.org/
[7] Elsevier ScienceDirect: http://www.sciencedirect.com/
[8] SpringerLink: http://link.springer.com/

2.2.8 CORE

CORE[9] (COnnecting REpositories) agreguje obsah různých celosvětových repozitářů a časopisů s otevřeným přístupem, obohacuje jej o metadata získaná analýzou textů za účelem poskytnout tyto informace široké veřejnosti. Klade velký důraz na opětovné použití dostupných materiálů, proto nabízí rozhraní API, které umožňuje další strojové využití předzpracovaných a obohacených dat.

Data jsou získávána z repozitářů pomocí k tomu určeného protokolu. Rozšířené vyhledávání se podobá ostatním službám podobného zaměření a je doplněno o fasetové vyhledávání, na jehož základě lze výsledky třídit podle typu publikace, jazyka, repozitáře, časopisu, roku vydání a autora. Výsledky jsou řazeny pouze podle relevance a jiný typ řazení není podporován.

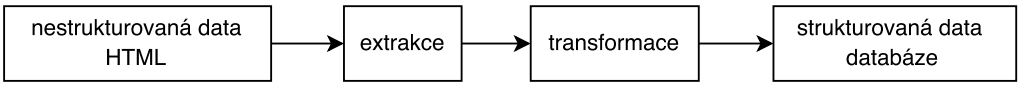
2.2.9 arXiv.org

arXiv.org[10] (X čteme jako řecké písmeno chí) je archiv vědeckých preprintů pod správou knihovny Cornellovy univerzity, který je v provozu od roku 1991. Zaměřuje se na matematiku, fyziku, nelineární systémy, informatiku, kvantitativní biologii a statistiku.

Grafické rozhraní působí velmi zastarale. Není podporováno žádné fasetové vyhledávání ani řazení výsledků. S dispozici je ale vyhovující pokročilé vyhledávání umožňující zadat klíčová slova, fráze i použít žolíkové znaky pro celou řadu atributů. Nalezené dokumenty lze stáhnout v různých formátech.

2.3 Web Scraping

Web scraping je technika, která slouží k získávání obecně nestrukturovaných informací z webových stránek, typicky v HTML formátu, a převádí je do strukturované podoby, aby mohla být opětovně použita. Zmíněný princip je vyobrazen na Obr. 2.1.



Obrázek 2.1: Obecný princip web scrapingu

2.3.1 Využívané techniky

Tato kapitola vychází z [12].

Data, převážně z HTML kódu webových stránek, lze extrahovat několika způsoby. V této kapitole uvádím několik z nich.

Tradiční kopírování a vkládání

Nejjednodušším a zároveň účinným způsobem web scrapingu je klasické ruční kopírování informací z jednoho místa na druhé (angl. *copy-and-paste*). Tento postup je sice náchylný na lidské chyby a při práci s rozsáhlými daty obtěžuje a unavuje, ale je nepostradatelný při dolování dat ze zdrojů, které se automatickým mechanismům explicitně brání.

[9] CORE: https://core.ac.uk/
[10] arXiv.org: http://arxiv.org/

Grepping a regulární výrazy

Pro potřeby portálů, které pracují s obrovským množstvím dat, je nezbytné celý proces úplně nebo alespoň z převážné míry zautomatizovat. Jednoduchý nástroj lze postavit na příkazu `grep`, který je součástí operačních systémů unixového typu nebo také pomocí *regulárních výrazů*. Tyto nástroje slouží k vyhledávání řetězců, které vyhovují požadovanému vzoru, a umožňují tak poměrně jednoduše extrahovat data, jejichž přibližná podoba je známa.

HyperText Markup Language (HTML) parser

Kategorie webů využívajících automatické generování obsahu má obvykle informace zakódovány v nějaké společné šabloně. *HTML parser* využívá faktu, že data mají částečně definovanou strukturu, kterou rozpozná *wrapper* a na základě identifikované šablony zvolí vhodný algoritmus, který vytěží příslušný obsah a převede ho do požadované podoby.

Document Object Model (DOM) parser

DOM parsování využívá DOM strom HTML dokumentu, ze kterého je možné informace rovněž dostat pomocí regulárních výrazů. Nicméně tento způsob se obecně nedoporučuje, protože je pomalý u rozsáhlejších dokumentů a také z důvodu, že přímo pro tyto typy dokumentů existují daleko lepší nástroje. Konkrétní položky stromové struktury lze rychleji a pohodlněji zpřístupnit prostřednictvím CSS selektorů nebo XPath.

2.3.2 Překážky extrakce obsahu

Klientské skriptovací jazyky

Klientské skripty jsou vykonávány na straně klienta, tj. za jejich správnou interpretaci je zodpovědný internetový prohlížeč. Nejrozšířenější technologie JavaScript a jQuery, knihovnu založenou na JavaScriptu, programátoři velmi často používají pro generování dynamického obsahu webu bez nutnosti obnovovat celou stránku. Pokud jsou použity tyto principy pro generování informací, není možné zmiňovaný obsah extrahovat tradičními metodami. Scraping by zde vedl pouze k získání zdrojového kódu stránky před vykonáním skriptu a očekávané informace, generované vykonáním skriptu, by chyběly. Spuštění JavaScriptu, například pomocí Pythonu, může být výpočetně náročné, výhodnější je proto hledat způsoby, kterými lze JavaScript parsovat přímo bez potřeby interpretace kódu [11].

CAPTCHA

CAPTCHA je test, který se používá pro odlišení člověka od robota. Spočívá obvykle v zobrazení automaticky generovaného obrázku s různě deformovaným textem, který spoléhá na schopnosti lidského mozku tento text správně rozpoznat, zatímco pro roboty nesmí být správně čitelný.

Smyslem takového způsobu ochrany je zabránit útočníkovi ve zneužití poskytovaných služeb prostřednicím webu. CAPTCHA zabraňuje robotům sloužícím útočníkům v odesílání zpráv, plnění internetových diskuzí spamy, dolování dat atd.

Robot, který chce překonat ochranu CAPTCHA, musí být schopen obrázek stáhnout, převést na text a odeslat serveru formulář se strojově získaným textem.

2.3.3 Požadavky na HTTP dotazy

Při zasílání požadavků je nutné si dát velký pozor a být zodpovědný. Mělo by být respektováno nastavení `robots.txt` a rozumný počet požadavků za jednotku času. Také by měl skript reagovat na stavová hlášení HTTP protokolu o případných chybách a vyvarovat se neúmyslnému odesílání formulářů, zejména těch přihlašovacích. V opačném případě může být toto jednání vyhodnoceno jako útok, který může mít pro útočníka nejrůznější důsledky.

Při implementaci web scraperu je chytré dbát na nastavení HTTP hlaviček, které se odesílají s každým dotazem na cílový server. Jejich nevhodné nastavení poskytuje prohledávanému serveru indície, které mohou účinně posloužit pro rozpoznání člověka od robota a vést až k případnému zablokování IP adresy a znemožnění tak stanovených cílů. Podle Mitchella [11] je základem robota podobajícího se člověku nastavení sedmi políček, se kterými většina webových prohlížečů inicializuje spojení. Nejdůležitější jsou především *User-Agent* a *Accept-Language*. Kompletní seznam zmiňovaných políček a jejich vzorové hodnoty lze nalézt v tabulce 2.1.

Host	https://www.google.com
Connection	keep-alive
Accept	text/html, application/xhtml+xml, application/xml; q=0.9, image/webp, */*; q=0.8
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/ 39.0.2171.95 Safari/537.36
Refferer	https://www.google.com
Accept-Encoding	gzip, deflate, sdch
Accept-Language	en-US,en;q=0.8

Tabulka 2.1: Příklad konfigurace HTTP hlaviček [11]

2.4 Elasticsearch

Data poskytovaná portálem je nutné uložit s ohledem na jejich efektivní a rychlé vyhledání. K tomuto účelu se pro své vlastnosti hodí *Elasticsearch*.

Jedná se o open-source fulltextový vyhledávač a bezeschémovou databázi, která je postavena na knihovně Apache Lucene[11]. Mezi jeho přednosti patří podpora horizontálního škálování možným přidáním dalšího clusteru v případě již nedostačujícího výkonu. Nejvíce se však pyšní svou rychlostí, díky které jsou výsledky při používání filtrů zobrazovány takřka okamžitě [7]. Podporuje funkce, jako jsou fasety, textové náhledy vyjadřující, v jakém kontextu byl daný výraz v dokument nalezen, hledání podobných dokumentů a další.

2.4.1 Analýza textu

Následující kapitola vychází z [14].

Před uložením dat do tzv. invertovaného indexu musí vstupní text projít analýzou, která se skládá z několika kroků.

[11] Apache Lucene: https://lucene.apache.org

Char filter

Char filter provede nalezení kořenů slov (hledání → hledat), expanzi slov na synonyma (hledat → hledat, pátrat, shánět), nalezení n-gramů (hledat → hle, led, eda, dat) a odstranění diakritiky.

Tokenizér

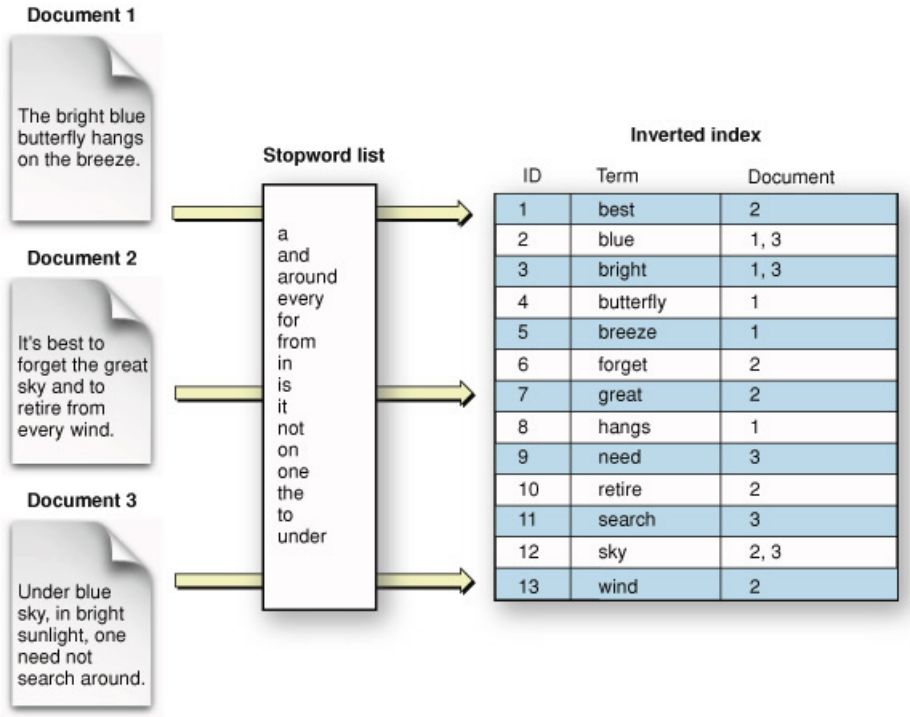
Tokenizér rozdělí vstupní řetězec na jednotlivá slova, tzv. tokeny nebo termy.

Token filtry

Token filtry nakonec zpracují jednotlivé tokeny, jejichž výsledkem je žádný, jeden nebo více tokenů. V této fázi jsou vyhledávána a zahazována tzv. *stopslova*. Jedná se o slova, která se opakují velmi často, například předložky nebo spojky, a proto nemají žádnou informační hodnotu. Současně se provádí *stematizace* (nalezení základu slova) a *lematizace* (určení základního tvaru slova).

2.4.2 Index

Takto zpracovaná data jsou následně pro potřeby rychlého fulltextového vyhledávání ukládána do invertovaného indexu. Invertovaný index je datová struktura, která umožňuje velmi rychle a efektivně nalézt indexovaná data za cenu zvýšení celkové velikosti databáze a režie na udržování indexu [5]. Název invertovaný index se odvozuje od faktu, že asociuje slova k dokumentům, ve kterých se dané slovo vyskytuje (viz Obr. 2.2).



Obrázek 2.2: Princip invertovaného indexu [1]

2.4.3 Fulltextové vyhledávání

Fulltextové vyhledávání je proces, který umožňuje rychle a efektivně vyhledat požadovaná klíčová slova v textu.

Vyhledávání může být realizováno vyplněním množiny polí, obvykle textových, nebo formulací jediného dotazu, který má podobu řetězce. Takový řetězec se obvykle skládá minimálně z klíčových slov. Pokud to syntaxe dotazovacího jazyka dovoluje, mohou být součástí dotazu navíc fráze, logické operátory (např. OR, AND, NOT), žolíkové znaky nahrazující jeden nebo více znaků, názvy konkrétních polí, kde mají být klíčová slova nalezena a další.

Široká podpora fulltextového vyhledávání je jeden z hlavních důvodů, proč bývá Elasticsearch volen jako databázový systém. Jinak tomu nebylo ani v mém případě.

2.4.4 Fasetové vyhledávání

Tato podkapitola vychází z [13].

Fasetové vyhledávání je spojením fulltextového vyhledávání a fasetové navigace. Uživatel nejdříve zadá dotaz a poté provede nad nalezenými výsledky konkretizaci pomocí fasetové navigace.

Fasetová navigace je organizování entit do několika navzájem disjunktních struktur, tzv. faset. Umožňuje efektivní prohledávání informačního prostoru. Každá faseta obsahuje množinu položek, které nazýváme fasetové hodnoty. V případě číselných hodnot je běžné mít možnost provést specifikaci pomocí zadání konkrétní numerické hodnoty nebo dokonce určením celého intervalu.

Vedle fasetové hodnoty, obvykle v závorce, bývá uveden počet entit, jejichž atribut má pravě tuto hodnotu. Hodnoty v rámci fasety jsou nejčastěji řazeny sestupně podle počtu entit. Mimo tento způsob se uplatňuje i abecední řazení. Uživatel může v rámci jedné fasety zvolit právě jednu nebo i vícero hodnot.

Elasticsearch přímo poskytuje prostředky k získání potřebných dat pro fasety a velmi usnadňuje jejich implementaci.

2.4.5 Rozhraní pro vyhledávání

Elasticsearch poskytuje pro vyhledávání velmi jednoduché rozhraní. Dotazy jsou jednoduše zapisovány ve formátu JSON a komunikace se serverem probíhá pomocí HTTP protokolu.

Dokumenty lze vyhledávat dvěma různými způsoby. Jejich rozdíl spočívá především ve výkonnosti a účelu, pro který je vhodné daný zápis použít.

Query

Klauzuli `query` je vhodné použít v případě, že nás zajímají skóre jednotlivých nalezených objektů. Tento typ dotazu z principu trvá déle kvůli režii spojené s výpočtem tohoto skóre. Dle něj jsou nakonec seřazeny nalezené výsledky a ty s nejvyšším skóre se nachází na prvních pozicích [6].

Příklad Vyhledání všech dokumentů, které obsahují v poli `longText` výraz `keyword` (pole `longText` představuje obecně dlouhý řetězec) [4]:

```
{
    "query": {
```

```
        "term": {"longText" : "keyword"}
    }
}
```

Výsledkem výše uvedeného výrazu může být například tato struktura [4]:

```
"hits" : {
    "total" : 48,
    "max_score" : 1.0,
    "hits" : [ {
        "_index" : "xstane34_projects",
        "_type" : "data",
        "_id" : "123456",
        "_score" : 0.4,
        "_source":{
            "field1": "value1",
            "field2": "value2",
            "longText": "This keyword will be found..."
            .
            .
            .
}
```

Ve výše uvedené struktuře jsou zajímavé především následující položky. Položka `total` vyjadřuje celkový počet nalezených dokumentů, `max_score` maximální hodnotu relevance vyhledávání k danému dotazu, `hits` obsahuje pole jednotlivých nalezených záznamů, `_score` představuje relevanci daného záznamu a pod klíčem `_source` se nachází konkrétní dokument, tak jak byl zaindexován a nalezen.

Filter

Při použití klauzule `filter` se skóre nepočítá, díky tomu je daleko rychlejší než `query`. Převážně se používá pro filtrování strukturovaných položek. Např. pokud nás zajímá, zda se datum pohybuje v uvedeném rozmezí nebo se jméno rovná konkrétní hodnotě atp. Často používané filtry mohou být uloženy v cache, čímž dochází k dalšímu zrychlení.

Příklad Vyhledání všech dokumentů, jejichž pole je právě rovno výrazu keyword [4]:

```
{
    "filter": {
        "term": {"value1" : "keyword"}
    }
}
```

Kapitola 3

Návrh a implementace systému

Na úvod je nutné říci, že vzorem navrhovaného a implementovaného systému je existující automaticky aktualizovaný webový portál o evropských výzkumných projektech (dále jen prototyp, viz kapitola 1.2). Ačkoliv byl původní kód, především extraktoru, z převážné části úplně přepsán, struktura portálu a také mnoho jiných původních myšlenek bylo zachováno. Z tohoto důvodu budu ve vhodných případech porovnávat předchozí a nový stav a poukazovat na to, k čemu vedly provedené změny.

Původní i nový systém je rozdělen na dvě základní logické jednotky – extraktor a portál, které pracují samostatně, ale dohromady tvoří jeden funkční celek. Tohoto členění se budu držet i v následujícím textu.

3.1 Extraktor dat

3.1.1 Analýza prototypu

Původní řešení extraktoru, které posloužilo jako prototyp pro nový vyvíjený extraktor, obsahovalo celou řadu nedostatků, které znemožňovavaly jeho správnou funkčnost. Aby bylo možné na něm postavit funkční systém a splnit požadavky zadání, musely být nejdříve odstraněny.

V této části nastiňuji pouze ty největší problémy, se kterými jsem se musel vypořádat. Jejich řešení je pak prezentováno dále v dalších kapitolách.

Chybné regulární výrazy

Od doby, kdy byla implementována část zajišťující extrakci požadovaných informací o projektu z její webové stránky, byla struktura HTML dokumentu změněna. Z tohoto důvodu již dávno nebyly platné některé regulární výrazy, které dotyčné položky získávaly.

Duplikace souhrnných zpráv

Souhrnné zprávy, které jsou součástí většiny projektů, byly exportovány ve formátu PDF, staženy a bylo k nim dále přistupováno stejně jako k vědeckým zprávám, které byly identifikovány unikátním hašem textové části dokumentu. Vzhledem k tomu, že souhrnné zprávy obsahují datum vytvoření PDF dokumentu (měnící se informace), byl při aktualizaci projektu haš pokaždé jiný a do systému byly ukládány opakovaně, jelikož byly považovány za nové.

Ignorace selhání extrakce

Původní extrakční skript identifikoval a vypisoval pouze základní chyby HTTP protokolu. Chyby tohoto rázu, pokud vedly k nezískání dat, byly sice vypisovány na příkazovou řádku, ale nebyly ani evidovány, ani nebylo možné jinak souhrnně zjistit, zda a které projekty nebyly úspěšně zaindexovány. Ostatní chyby nebyly uvažovány vůbec a vedly k masovému ztrácení projektů. V některých případech končily neošetřenou výjimkou a okamžitým ukončením skriptu.

3.1.2 Analýza zdroje informací

Informace o projektech jsou získávány z velmi nespolehlivého zdroje, webového portálu Cordis, který je nespolehlivý hned z několika důvodů.

Tato kapitola má v tomto textu své místo, protože právě problémy se zdrojem dat mne provázely po celou dobu mé práce, vedly k razantnímu přepsání původního kódu prototypu a stály mne mnoho úsilí a námahy.

Nestálé a neúplné informace o projektu

Webová stránka s informacemi o projektu, tak jak ji zná uživatel portálu Cordis, generuje informace do stále stejné šablony. Nicméně i přesto byla tato struktura od doby návrhu prototypu změněna, pravděpodobně se změnou designu portálu Cordis. Tato stránka také z neznámého důvodu v některých případech neobsahuje odkaz na oficiální webovou stránku projektu, ačkoliv v exportovaném XML dokumentu je jasně uvedena.

Z tohoto důvodu jsem přistoupil na extrahování informací o projektu z XML dokumentu namísto z HTML stránky. Tím byl vyřešen zmíněný problém a současně jsem získal přístup k datům v lepší formě, jednodušším a přehlednějším způsobem pomocí XPath.

Odezva portálu

Portál Cordis má na první pohled velmi pomalou odezvu, a to zejména při vyhledávání, a tak byla extrakce velmi pomalá a zdlouhavá.

Proto jsem se rozhodl provádět paralelní stahování stránek se seznamem projektů, XML dokumentů se samotnými projekty a také vědeckých zpráv a souhrnných zpráv. Tímto krokem bylo dosaženo obrovského zrychlení (viz kapitola 4.2).

Chybějící projekty

Aby se extraktor dozvěděl o existenci projektů, musí systematicky prohledávat Cordis a z nalezených výsledků pak extrahovat jejich unikátní identifikátor (tzv. record number projektu). Bohužel se velmi často stává, že v seznamu nalezených projektů není tolik projektů, kolik deklaruje počet nalezených výsledků vyhledávání (nemluvě o případech, kdy na stránce nejsou výsledky žádné). Obdobným problémem je také občasné chybové hlášení serveru o nemožnosti provést vyhledávání nebo nevalidní XML dokument, který obsahuje z neznámého důvodu HTTP hlavičku.

Proto byla navržena robustní struktura extraktoru, která nedá možnost ztrátám projektů.

3.1.3 Požadavky

Extraktor je navrhován jako konzolová aplikace. Kvůli omezeným možnostem textového výstupu musí vypisované informace dostatečně a přehledně informovat uživatele o průběhu extrakce a o neočekávaných událostech.

Má za úkol spolehlivě a zodpovědně nalézt všechny projekty na informačním portálu pro evropský výzkum a inovace Cordis[1]. Z nalezených projektů je nezbytné následně extrahovat obecné informace o projektu včetně webové stránky projektu a URL odkazů na vědecké a souhrnné zprávy a ty následně stáhnout. Pokud není web projektu uveden, je automaticky vyhledán pomocí internetového vyhledávače. V případě nalezení webu projektu jsou z něj získány další vědecké zprávy. Na závěr se získané projekty a vědecké zprávy zaindexují do databáze. Z důvodu informování administrátora o případných problémech je potřeba evidovat všechny neočekávané události tak, aby mohly být interpretovány přehledným a účelným způsobem.

Dále musí umět nalézt již existující projekty a reagovat na nové nebo upravené projekty na Cordis, podporovat extrahování uživatelem specifikovaných projektů a zvládnout aktualizovat případné nové vědecké zprávy z webových stránek projektů.

Extraktor také musí respektovat přednost změn provedeným uživatelem.

3.1.4 Použité technologie

Pro implementaci byl zvolen skriptovací jazyk Python, který se osvědčil už při implementaci prototypu. Byl zvolen především z toho důvodu, že nabízí velmi dobré rozhraní pro práci s databází Elasticsearch (knihovna `elasticsearch-py`).

Extrakce vědeckých zpráv z webových stránek projektů byla provedena využitím nástroje `RRSDeliverables` vyvinutým skupinou ReReSearch na FIT VUT v Brně [9].

Pro extrakci prostého textu z vědeckých zpráv ve formátu PDF je použit nástroj `pdftotext` a pro extrakci formátů DOC a DOCX `soffice`, protože oba nástroje poskytovaly velmi dobré výsledky.

Mezi stěžejní knihovny patří `elasticsearch-py`, která umožnila jednoduchou práci s databází, na knihovně `xml.etree.ElementTree` je postaven extraktor XML dokumentů, knihovna `urllib2` byla použita pro veškerou komunikaci prostřednictvím sítě (stahování i ověření dostupnosti URL adresy) a pomocí `multiprocessing` jsem implementoval paralelní operace.

3.1.5 Uživatelské rozhraní

Vzhledem k tomu, že je extraktor konzolová aplikace, byla vytvořena sada několika příkazů, která slouží k jeho ovládání. V následující části bude předvedena práce s extraktorem. Příkazy nejsou uvedeny obecně, ale pro maximální názornost jsou zapsány se vzorovými hodnotami parametrů.

Extraktor slouží jak k jednorázovému naplnění databáze, tak k pravidelným aktualizacím. Z tohoto důvodu extraktor podporuje několik následujících příkazů. Každý příklad příkazu je uveden stručným popisem jeho funkce.

Zaindexování projektů z Cordis s datem startu v zadaném rozsahu:

```
python extractor.py -u "01/01/2016""31/12/2016"
```

[1] Cordis: http://cordis.europa.eu

Zaindexování projektů ze seznamu v souboru:

```
python extractor.py -f seznamProjektu.txt
```

Zaindexování aktualizovaných a nových projektů od data poslední aktualizace:

```
python extractor.py -n
```

Aktualizace vědeckých zpráv z webu projektu u projektů s datem začátku v uvedeném rozmezí:

```
python extractor.py -e "01/01/2013""31/12/2013"
```

Nalezení webu projektů vyhledávačem a stažení vědeckých zpráv:

```
python extractor.py -y
```

3.1.6 Informování o průběhu extrakce a evidování událostí

Informování o průběhu extrakce

Extraktor byl v oblasti informování uživatele o průběhu extrakce a neočekávaných stavech, které během ní mohou nastat, v porovnání s prototypem značně vylepšen. Uživatel, mimo jiné, ani neměl představu o tom, jak extrakce postupuje a jak dlouho by mohla trvat.

Bylo implementováno několik triviálních funkcí pro výpis chyb, varování, úspěšných událostí, pojmenování činností a další. Tyto funkce se liší pouze tím, jakým způsobem formátují vypisovaný text v příkazové řádce (barva textu a pozadí) a slouží spíše pro přehlednost kódu a jednoduchost případných změn zmíněného formátování.

Uživatel je prostřednictvím těchto hlášení informován o tom, v jakém režimu extraktor běží, kolik stránek se seznamy projektů bude extrahováno, kolik projektů bude staženo, jaký projekt je aktuálně zpracováván, kolik vědeckých a souhrnných zpráv z jakých zdrojů bude staženo. Po dokončení každé dílčí operace (tj. extrakce seznamů s projekty, stažení projektů, stažení vědeckých a souhrnných zpráv z daného zdroje nebo zaindexování projektu) je vypsán počet úspěšných a neúspěšných podoperací. Každá podoperace je navíc doplněna o její pořadové číslo vztažené k celkovému počtu podoperací v rámci dané operace, aby měl uživatel představu, v jaké fázi se extrakce nachází (např. PROJ(1/160)). Po skončení každé operace je vypsána doba celkového běhu skriptu. Některé výpisy jsou navíc rozšířeny o procentuální vyjádření pokroku.

Zmíněná hlášení byla navržena s ohledem na multilingválnost. Znění všech hlášení může být jednoduše přeloženo do jiného jazyka (případně jinak modifikováno) na jednom jediném místě v modulu `extractor.config.lang`.

Všechna uvedená hlášení jsou vypsána okamžitě, jakmile je vyvolána navázaná událost, což se liší od evidování událostí.

Evidování událostí

Evidování událostí se skládá z logování informací do textového souboru a ukládání informací do vymezených položek v JSON struktuře daného projektu (zprávy), která je spolu s informacemi o projektu (zprávě) zaindexována do databáze.

V případě logování jsou ukládány jen nezbytné informace, které mají pro administrátora nějaký význam. Konkrétně se jedná o hlášení o chybách a úspěšnosti zaindexování projektu včetně souhrnné statistiky a době běhu skriptu. Logy jsou seskupeny v adresářích podle data

aktualizace. Hlášení nejsou do souboru ukládána okamžitě, ale po ukončení dané skupiny paralelních procesů v rámci jedné operace, protože souběžně běžící procesy by způsobovaly v logu chaos.

Některé informace, ze kterých lze zpětně interpretovat problémy při extrakci, jsou ukládány také přímo s projektem, případně zprávou do databáze. Jedná se o chybový stavový kód HTTP protokolu vyvolaný při načítáni webové stránky projektu a chyby vzniklé při zpracování zprávy (nelze stáhnout dokument nebo extrahovat text z dokumentu). Tyto pomocné položky poskytují cenná data pro výpočet statistik o stavu projektů a zpráv v databázi.

3.1.7 Paralelní zpracování

Vhledem k tomu, že získávání informací z portál Cordis, odkud jsou stahovány informace o projektech a zprávy, je značně pomalé, bylo stahování a zpracování těchto informací implementováno tak, aby probíhalo paralelně. Současně napomohlo i rychlejšímu zpracování vědeckých zpráv stahovaných z webových stránek projektů.

Uváděnou funkcionalitu implementuje s použitím knihovny `multiprocessing` funkce `getParalelURLs()` v modulu `extractor.common`, které je předán název funkce vykonávané paralelně, seznam parametrů této funkce pro každý dílčí proces a maximální počet souběžných procesů.

V případě zpracování XML seznamu projektů nebo konkrétního projektu je zmíněné funkci parametrem předána funkce `getURLXMLRoot()`, která vrací kořenový element XML dokumentu.

Pokud je zpracovávána vědecká zpráva, je funkci předána funkce `getDelivFile()`, respektive `getReportXML()` v případě souhrnných zpráv.

3.1.8 Vyhledání projektů

Před procesem samotné extrakce konkrétních projektů musí být nejdříve získán seznam jejich jednoznačných identifikátorů, aby extraktor projektů věděl, jaké projekty má stáhnout, zpracovat a nakonec zaindexovat do databáze. Tento seznam může být buď předán extraktoru formou souboru, nebo být vytvořen automaticky. Druhá varianta bude dále rozebrána v této části.

findProjects()

K nalezení projektů slouží funkce `findProjects()` v modulu `extractor.extractor`, které je předáno rozmezí dat začátku projektů (neaktualizační režim), respektive dat poslední aktualizace na Cordis (aktualizační režim). Dalším parametrem je maska URL adresy sloužící k procházení výsledků vyhledávání na portálu Cordis. Dle tvaru adresy je rozlišen aktualizační a neaktualizační režim funkce. Poslední parametr specifikuje umístění systémového souboru, kam je uložen získaný seznam identifikátorů projektů.

Funkce `findProjects()` rozdělí interval dat na menší podintervaly dlouhé maximálně jeden rok, kvůli potřebě redukovat počet nalezených projektů (Cordis umožňuje nalézt maximálně 5010 projektů).

Následně je pro každý interval zavolána dvojice funkcí `getURLsInInterval()` a `getRCNs()`, která poskytne seznam identifikátorů projektů. Ten je nakonec uložen do systémového souboru `project_rcns.txt`.

getURLsInInterval()

Funkce `getURLsInInterval()` nejprve zjistí celkový počet nalezených výsledků (očekávaný počet projektů pro zpracování) a poté na základě něj vygeneruje seznam URL adres vedoucích na každou stránku s výsledky spolu s očekávaným počtem projektů na dané stránce (ve své podstatě zajišťuje stránkování výsledků).

getRCNs()

Funkce `getRCNs()` pak převezme seznam URL adres s počtem očekávaných projektů na každé stránce a tyto adresy využije pro vytvoření seznamu parametrů, který je potřebný pro paralelní zpracování (viz kapitola 3.1.7). Nalezené identifikátory projektů získané ze stránek s výsledky jsou vráceny funkcí ve formě seznamu. Pokud některou stránku není možné zpracovat kvůli problému na straně portálu (viz odstavec Chybějící projekty v 3.1.2) nebo HTTP chybě, je odkaz na selhanou stránku uložen do systémového souboru `project_pages.txt` a při každém dalším volání `findProjects()` se skript implicitně opět pokusí uložené dříve selhané seznamy zpracovat.

3.1.9 Zpracování projektů

Projekty, které mají být zaneseny do systému, jsou dány seznamem jejich identifikátorů. Pokud extraktor běží v aktualizačním režimu, je jako seznam implicitně uvažován systémový soubor `project_rcns.txt`, v opačném případě musí být zdrojový soubor specifikován uživatelem.

indexProjects()

Za zpracování projektů je zodpovědná funkce `indexProjects`, která je součástí modulu `extractor.extrctor`. Této funkci je parametrem předán soubor obsahující seznam identifikátorů projektů pro zpracování. Jednotlivé identifikátory jsou postupně transformovány na URL adresy, které jednoznačně identifikují projekt na portálu Cordis. Z těchto adres je poté vytvořen seznam parametrů pro funkci zajišťující paralelní stažení XML dokumentů projektů (viz kapitola 3.1.7). Výsledkem je seznam kořenových elementů všech XML souborů s informacemi o projektech.

indexProject()

Seznam kořenových elementů je následně iterován a pro každý element je volána funkce `indexProject()`. Funkce nejdříve vytvoří novou instanci třídy `Project`, která je inicializována kořenovým elementem projektu. Zavoláním metody `fillData()` dojde k naplnění atributů třídy vyextrahovanými položkami projektu a po vyvolání metody `indexData()` je projekt včetně nalezených zpráv zaindexován do databáze.

fillData()

Metoda `fillData()` nejdříve extrahuje pomocí opakovaného aplikování funkce `parseRecord()` všechny položky projektu, které jsou ze stromové struktury XML dokumentu jednoduše zpřístupněny pomocí adresy XPath. Pokud je nalezena adresa web projektu, je ověřena její dostupnost a následně je uložena spolu se stavovým HTTP kódem do struktury JSON reprezentující informace o webové stránce projektu.

findDeliverables()

Pokud není doposud známa adresa podstránky webu projektu s vědeckými zprávami, skript se ji pokusí nejdříve určit pomocí funkce `findDeliverables()`, které je tato adresa (obvykle domovské stránky) předána. Samotné extrahování zpráv je dále popsáno v samostatné kapitole 3.1.10.

indexData()

Metoda `indexData()` nejdříve ověří, zda se indexovaný projekt již nachází v databázi. Pokud ne, projekt se zaindexuje jako nový s aktuálním časem indexace. V opačném případě dojde k jeho aktualizaci s aktuálním časem aktualizace.

Na závěr je zavolána metoda `indexDeliverables()` z modulu `extractor.delivs`, která zaindexuje vědecké a souhrnné zprávy předané ve formě seznamu (podrobně v 3.1.10).

parseRecord()

Metoda `parseRecord()` je jednou z nejdůležitějších funkcí extraktoru. Slouží k extrakci hodnot elementů z XML dokumentu. Adresování strukturovaných i nestrukturovaných elementů se provádí pomocí adresy XPath dosazené za parametr `path`.

V případě adresování strukturovaného elementu (např. koordinátor, který má svůj název, zemi, město, jméno, příjmení, ...) se za parametr `path` dosadí adresa rodičovského elementu, společného všem podelementům, které chceme získat. Parametr `subPaths` umožňuje definovat seznam adres, relativních k rodičovskému elementu, který specifikuje umístění požadovaných podelementů a také umožňuje zvolit název dané položky ve výsledné struktuře. Parametr `normalize` povolí normalizaci textových řetězců (např. odstranění přebytečných bílých znaků, spojení rozdělených slov, odstranění HTML a XML tagů atd.)

Funkce vrátí extrahované informace ve formátu JSON, který je navržen tak, aby byl přímo použitelný pro uložení do databáze.

Příklad extrahování téma projektu:

```
parseRecord(
  xmlRoot=xr,
  multiple=True,\
  path='associations/programme[@type="relatedSubProgramme"]',
  subPaths=(('code', 'code_oldName'),
            ('title', 'title_oldName')),
  normalize=True)
```

3.1.10 Zpracování zpráv

Logika kolem zpráv, tedy vědeckých a souhrnných zprávy je velmi složitá a na první pohled není vidět. Z tohoto důvodu tuto problematiku popisuji podrobně. Tento popis výrazně usnadní případné budoucí zásahy do kódu.

Následující výčet srovnává základní fakta:

- Vědecká zpráva
  - získáno z Cordis (Documents and Publications) nebo webu projektu

  - jakýkoliv dokument, obvykle PDF, DOC nebo DOCX
  - originál i prostý text uložen v souborovém systému
  - prostý text uložen v DB, pokud PDF, DOC nebo DOCX a převod úspěšný
  - jednoznačnost na základě haše binárního souboru
- Souhrnná zpráva
  - získáno z Cordis (Report Summaries)
  - XML dokument
  - uložen pouze prostý text v souborovém systému
  - prostý text uložen v DB vždy
  - jedinečnost na základě unikátního identifikátoru zprávy

Pokud není možné zprávu stáhnout, není uložena do souborového systému a je uložena do databáze bez textu a jednoznačného identifikátoru (není myšlena položka `_id`). Pokud není možné získat text vědecké zprávy, je i přesto uložena do databáze.

findDeliverables()

Metoda `findDeliverables` očekává jako parametr URL adresu webu projektu, pak vrací URL adresu podstránky, kde se nacházejí vědecké zprávy, a seznam URL adres nalezených zpráv a jejich titulků. Pokud je funkci předána přímo adresa podstránky, je vrácen pouze zmíněný seznam.

parseDeliverabes()

Odkazy na vědecké zprávy z Cordis, vědecké zprávy z webu projektu a souhrnné zprávy z Cordis jsou odděleně předány funkci `parseDeliverabes()`, která zprávy paralelně stáhne, u vědeckých zpráv určí haš souboru, ze souborů ve formátu PDF, DOC a DOCX extrahuje text pomocí nástrojů `pdftotext` a soffice. U souhrnných zpráv je za haš považován jejich unikátní identifikátor a jejich text se extrahuje přímo z jejich XML dokumentu. Textový obsah obou typů zpráv je uložen ve formátu textového souboru do souborového systému.

indexDeliverables()

Na závěr jsou zprávy identifikované hašem, titulkem a URL adresou zdroje předány funkci `indexDeliverables()`, která postupně načte jejich textový obsah ze souborového systému a zaindexuje je do databáze. Zprávy, které nebylo možné stáhnout, jsou v databázi uloženy bez haše a ty, které nebylo možné převést, jsou evidovány bez textu. Pokud je v databázi nějaká zpráva, která nebyla stažena nebo nebyl extrahován její text, a při některé další aktualizaci je zpráva opět plně dostupná, je v databázi aktualizován původní záznam. Naopak je databáze ošetřena tak, že úspěšně uložené zprávy nemohu být při aktualizaci smazány či jinak znehodnoceny.

3.1.11 Vyhledávání stránek projektů

Extrahování vědeckých zpráv z webových stránek projektu řeší kapitola 3.1.10. Problém nastává, pokud odkaz na tuto stránku není uveden na portálu Cordis, ačkoliv web existuje. Aby portál, který je předmětem této práce, poskytoval přístup ještě k většímu počtu vědeckých zpráv, je možné webovou stránku projektu vyhledat automaticky za pomoci internetového vyhledávače a poskytnou ji extraktoru.

Řešení s Googlem

Jako první se nabízel vyhledávač Google, který má ale velmi tvrdá pravidla související s používáním robotů a počtem požadavků za jednotku času. Experimentálně bylo zjištěno, že je možné provést kolem padesáti požadavků za sebou a poté je nezbytné více než hodinové čekání, aby ochrana CAPTCHA zmizela a bylo možné pokračovat.

Řešení s Yahoo!

Snadnější řešení nabídl vyhledávač Yahoo!, který dovoluje až tisíc požadavků za hodinu, což naprosto dostačuje tempu, kterým jsou webové stránky projektů prohledávány.

Implementace

Pro práci s vyhledávačem byla naimplementována v modulu `extractor.common` velmi jednoduchá funkce `searchOnYahoo()` postavená na odlehčené mobilní verzi vyhledávače. Funkce očekává jako parametr akronym projektu (značen jako `<A>`), který je využit pro vytvoření dotazu:

```
+eu +project "<A>"(inurl:"<A>"OR intitle:"<A>") -site:europa.eu site:eu
```

Bylo rovněž experimentováno s verzí dotazu, ve kterém byl uveden i titulek projektu. Avšak přidání titulku projektu zamezilo u převážné většiny projektů ze zkoumaného vzorku nalezení webové stránky.

Funkce `searchOnYahoo()` na závěr vytvořený dotaz odešle na server a ze získané odpovědi pak pomocí regulárních výrazů vyextrahuje URL adresy nalezených stránek a ty pak vrátí.

Zmíněná funkce je pak použita v modulu `extractor.project` ve funkci `searchWebsite()`, která vrátí adresu nalezené stránky ve formátu vhodném pro uložení do databáze. Adresa je vrácena pouze v případě, že její generická doména je rovna „eu“ a v doméně se nachází podřetězec akronymu (bez ohledu na velikost písmen a přítomnost případných nealfanumerických znaků).

Navržená podmínka sice nepropustí mnoho relevantních webů vzhledem k výše uvedenému dotazu, ale na druhou stranu zabrání zahlcení databáze nesmyslnými adresami, které by musel administrátor zdlouhavě ověřovat a schvalovat. Nutno zdůraznit, že i webový portál poskytuje rozhraní pro vyhledávání a doplňování webových stránek projektů.

Pro doplnění projektů o webovou stránku z vyhledávače je nutné extraktor spustit ve speciálním režimu (viz kapitola 3.1.5). Pokud skript nalezl web projektu nebo se o to alespoň pokusil, je projektu nastaven příznak, který zajistí, že se daný projekt nebude nadále nabízet tomuto skriptu.

3.2 Webový portál

3.2.1 Požadavky

Webový portál má za úkol poskytnout přehledné a praktické webové rozhraní pro vyhledání projektů a zpráv v databázi, zobrazování nalezených projektů, přehled statistik o stavu dat v databázi a administrační část sloužící pro řešení problémů s daty. Webový portál musí být rozdělen jak na sekci přístupnou pro všechny uživatele, tak na administrátorskou část.

Vyhledávání musí být možné realizovat kombinací klíčových slov a standardní fasetové navigace, přičemž pro rychlou orientaci v položkách faset je potřebná podpora filtrování těchto položek. Výpis výsledků musí být přehledný a umožňovat pohodlné stránkování.

Výpis informací o projektu musí nabízet maximum možných informací a podporovat stažení projektových zpráv z lokálního souborového systému pro případ, že by byla zpráva z původního umístění odstraněna.

Administrační část musí zprostředkovávat informace o proběhlých aktualizacích, umožnit přístup k existujícím logům, zobrazovat problémy s webovými stránkami, poskytnout prostřední pro schvalování automaticky přiřazených adres webových stránek a podporovat úpravy problematických položek projektů.

Přehled statistik musí nabízet dostatek statistických údajů, aby si administrátor mohl udělat dostatečnou představu o stavu projektů a zpráv v databázi.

3.2.2 Použité technologie

Pro implementaci webového portálu byl rovněž použit skriptovací jazyk Python. Díky této volbě bylo možné pro oba moduly implementovat společné konfigurační soubory a také mohly být do modulu `portal` importovány některé funkcionality z modulu `extractor`, které se uplatnili zejména při editaci projektů prostřednictvím webového portálu.

Pro práci s databází byla použita opět knihovna `elasticsearch-py` a navíc i knihovna `elasticutils`. Webový portál byl vyvinut pomocí frameworku Flask[2], Bootstrap[3] a šablonovacího systému Jinja2[4].

3.2.3 Uživatelské rozhraní portálu

Vyhledávání

Podoba stránky pro vyhledávání byla upravena jak vizuálně, tak funkčně (viz Obr. 3.1).

Výčet vylepšení:

- logo, počet výsledků, přepínání mezi projekty a zprávami, podoba pravého boxu, formátování výsledků vyhledávání, stránkování
- přesunuty fasety a kompletně přepracována jejich podoba a funkce
- seskupování stejných zpráv k jednomu projektu bez opakování jeho názvu
- možnost zobrazit statistiky pro zadaný dotaz
- globální horizontální navigace

[2] Flask: http://flask.pocoo.org/
[3] Bootstrap: http://getbootstrap.com/components/
[4] Jinja2: http://jinja.pocoo.org/

Obrázek 3.1: Screenshot stránky pro vyhledávání

3.2.4 Vyhledávání

Vyhledávání je založeno na fasetovém vyhledávání, které se skládá z fulltextového vyhledávání a fasetové navigace. Uživatel má možnost zadat do vyhledávacího pole vyhledávače požadovaná klíčová slova a nalezené výsledky dále upřesnit pomocí fasetové navigace v levé části stránky. Pokročilejší uživatel může rovnou formulovat pomocí dotazovacího jazyka i složitější dotazy, než je schopen vytvořit pomocí uživatelského rozhraní.

Dotazovací jazyk

Syntaktický a sémantický analyzátor pro zpracování dotazovacího jazyka byl implementován pomocí nástroje PLY[5]. Pomocí tohoto analyzátoru se z dotazu získává struktura hodnot faset a seznam klíčových slov. Na základě těchto získaných informací fasetová navigace ví, které položky má zaškrtnou a které ne.

[5]PLY: http://www.dabeaz.com/ply/

Fasetová navigace

Titulky faset, pořadí a jejich označení v dotazovacím jazyku je dáno jednoduchým řazeným slovníkem v modulu `portal.search`, který umožňuje přehledným a rychlým způsobem fasety jakkoliv modifikovat.

Při výpočtu hodnot pro všechny fasety, viditelné na portálu po levé straně, jsou postupně iterovány jednotlivé položky této struktury a na základě uvedených informací je vytvořeno pole obsahující titulek fasetu, označení fasetu v jazyku a seznam fasetových hodnot a počet odpovídajících dokumentů. Data pro fasety jsou získány pomocí funkce `facet()` z knihovny `elasticutils`. Toto pole je pak následně vypsáno na webovém portálu.

Uživatel může v rámci jedné fasety zvolit více položek, mezi kterými je implicitně uvažována disjunkce. Po odeslání požadovaného nastavení faset je jejich nastavení převedeno do syntaxe dotazovacího jazyka, vyhodnoceno a uživateli se zobrazí požadované výsledky.

Vzhledem k tomu, že použitý dotazovací jazyk je podmnožinou jazyka, který využívá `Query String Query`, může být dotaz přímo vyhodnocen databází Elastic. Dotazovací jazyk podporuje klíčová slova, fráze, porovnávací operátory pro specifikaci rozsahu hodnot, logické operátory, žolíkový znaky „?“ a „*“ a seskupování pomocí závorek.

Příklad složitějšího dotazu

```
management partcountry:(Germany AND "United Kingdom")
coordinator:*university* year:>=2015 programme:(FP7 H2020)
```

Pod fasety, které obsahují více než 6 položek (výjimka rok), je zobrazeno vyhledávací pole pro filtrování výsledků. Při každé změně obsahu pole se zahájí komunikaci s databází pomocí jQuery, která vrátí pouze ty fasetové hodnoty, v kterých je obsažen zadaný podřetězec.

Vyhledání v databázi

Dotaz, který je zaslán databázi pro vyhledání požadovaných projektů, má níže uvedenou strukturu. Jak již bylo zmíněno, klíčová slova a hodnoty faset jsou zpracovávány odděleně. Klíčová slova jsou vyhledávána v pevně definovaných položkách, přičemž největší váhu má `acronym`. Hodnoty faset jsou vyhledávány v těch položkách, které jsou dány dotazovacím jazykem.

Struktura používaného dotazu pro vyhledávání projektů v databázi:

```
{"query":
  {"bool":
    {"must": [
      {"query_string": {
        "default_operator": "and",
        "fields": ["acronym^6", "title^5",
                   "objective^3", "fundedUnder.subprogramme^2",
                   "website.origWeb"],
        "query": "project"}
      },
      {"query_string": {
        "default_operator": "or",
```

```
          "query": "fundedUnder.programme:(fp7)"
        }}]}}
    }
```

3.2.5 Stránka projektu

Stránka, která slouží k zobrazení informací o projektu, byla v několika ohledech vylepšena oproti původnímu řešení:

- nové položky – stav projektu, datum aktualizace na Cordis i v databázi, adresy účastníků
- změna designu
- stahování zpráv z lokálního souborového systému
- informace o typu souboru zprávy, v po najetí myší na zprávu se zobrazují alternativní názvy (pouze u duplicitních zpráv)

3.2.6 Administrace

Přihlašování uživatelů

Administrační část je přístupná pouze pro přihlášené uživatele. Autentizace uživatelů je implementována v modulu `portal.authentication`. Existující uživatelé jsou definováni pouze zápisem ve zdrojovém kódu a nejsou implementovány prostředky pro registraci uživatelů prostřednictvím portálu. Nicméně modul je navržen tak, že jej lze jednoduše napojit na databázi.

Informace o aktualizacích

Podstránka s aktualizacemi zobrazuje seznam dat, kdy proběhla nějaká změna databáze. Tyto hodnoty jsou odvozené od pomocných informací ukládaných spolu s projekty do databáze. Po kliknutí na zvolené datum, je zobrazen případný seznam logů proběhlých aktualizací v tento den. Tyto logy byly podrobně rozebrány v kapitole 3.1.6. Textový obsah souboru logu je pomocí regulárních výrazů obarven tak, aby byl log přehlednější. Pod výpisem logů je tabulka v tento den zaindexovaných projektů a aktualizovaných projektů.

Problémy s webovými stránkami

Podstránka s problémy obsahuje několik tabulek, které poskytují přístup k projektům, které mají nějaký problém s webovou stránkou. Součástí každé tabulky je tlačítko, které umožňuje daný projekt modifikovat a problém vyřešit.

U projektů, jejichž webová stránka byla zjištěna automaticky pomocí vyhledávače, se nachází tlačítko, které slouží k potvrzení toho, zda byla přiřazena správná webová stránka. Některé webové stránky byly tzv. potvrzeny automaticky, jelikož na nich byly nalezeny nějaké vědecké zprávy. Každý řádek tabulky také obsahuje tlačítko s ikonou lupy, která slouží k účelu, proč byla tato sekce implementována – oprava chybně automaticky přiřazených nebo vůbec nenalezených dat.

3.2.7 Úprava projektu

Při editaci projektů je možné měnit a mazat webové stránky a vědecké zprávy. Logika této stránky je navržena tak, že při změně URL webové stránky je tato stránka prohledána a pokud jsou nalezeny nové vědecké zprávy, jsou zaindexovány. Pokud je změněna URL adresa vědecké zprávy, je původní vědecká zpráva smazána a místo ní je uložena zpráva z nové adresy.

3.2.8 Statistiky

Statistiky jsou novou sekcí portálu. Pro přístup do této sekce musí být uživatel přihlášen. Využívají celkem 23 triviálních různých dotazů, které poskytují velmi podrobný náhled na stav databáze. Statistky je možné před jejich výpočtem filtrovat a získat tak přehled o požadované skupině projektů.

3.3 Ukládání dat

Data o projektech a zprávách jsou ukládána do databáze Elasticsearch, která byla zvolena, protože nabízí velmi rychlé operace s velkým množstvím dat a podporu fulltextového vyhledávání.

Informace jsou ukládány celkem do dvou indexů – zvlášť projekty a vědecké zprávy.

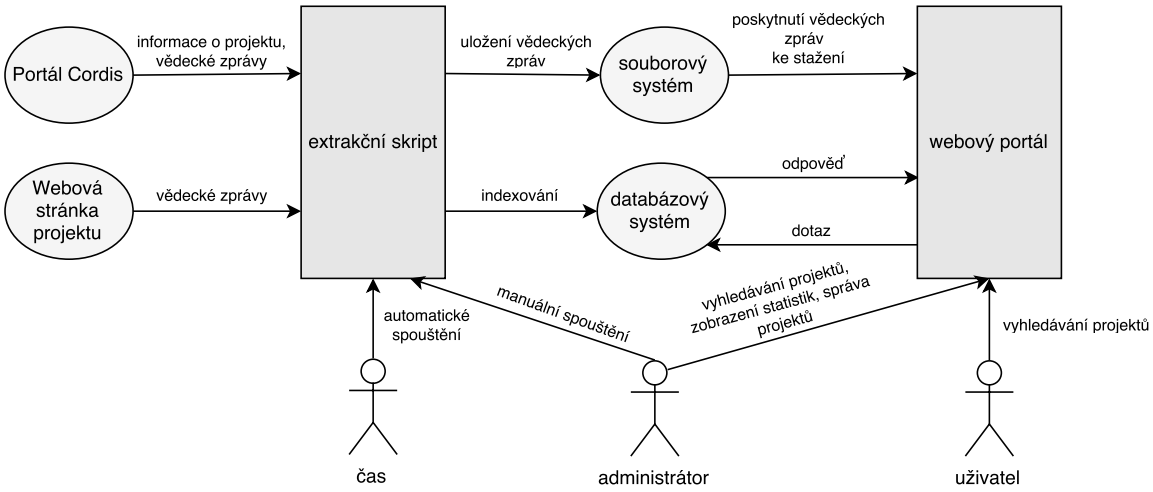
Původní struktura databáze byla z převážné většiny přepracována a velké množství položek je strukturováno jako vnořené položky, které zajišťují především větší přehlednost při práci s databází.

Příklad struktury databáze s vnořenými položkami:

```
'coordinator': {
    'type': 'nested',
    'include_in_parent': True,
    'properties': {
        'name':        { 'type': 'string', 'analyzer': 'analyzer_keyword' },
        'shortName':   { 'type': 'string', 'analyzer': 'analyzer_keyword' },
        'country'      { 'type': 'string', 'analyzer': 'analyzer_keyword' },
        'address':     { 'type': 'string', 'analyzer': 'analyzer_keyword' }
        .
        .
        .
                    }}
```

3.4 Schéma systému

Na obr. 3.2 je vyobrazeno schéma celého systému, které vyjadřuje, jak se sebou jednotlivé části souvisí a interagují.



Obrázek 3.2: Schéma systému

Kapitola 4

Vyhodnocení systému

4.1 Získaná data

Vyvinutý systém automaticky indexuje průměrně cca 550 nových a 4 500 aktualizovaných projektů z portálu Cordis za měsíc. Z tabulek 4.1 a 4.2 vyplývá, že bylo k dnešnímu dni úspěšně získáno 33 050 projektů, 7 621 vědeckých zpráv z Cordis, 10 405 vědeckých zpráv z projektových stránek a 32808 souhrnných zpráv, které jsou k dispozici v databázi. 3 675 projektů má projektovou stránku, přičemž u dalších 6 707 projektů byla webová stránka automaticky dohledána.

	Vědecké zprávy z Cordis	Vědecké zprávy z webu projektu	Souhrnné zprávy
Nalezeno odkazů	8 027	13 416	32 810
Úspěšně zaindexováno	7 621	10 405	32 808
Nelze stáhnout	129	2 240	2
Nelze získat prostý text	277	771	0

Tabulka 4.1: Statistika zpráv

Počet projektů	Hodnota
Celkem	33 050
S webem	10 382
S dostupným webem	9 278
Se zprávami z webu	979
S dohledaným webem	6 707
Se zprávami z dohledaného webu	677

Tabulka 4.2: Statistika projektů s webovou stránkou

4.2 Rychlost

Systém byl oproti prototypu značně urychlen použitím paralelních procesů na několika úrovních zpracování dat. Jako zkoumaný vzorek dat byla zvolena množina projektů, která má různé počty vědeckých zpráv z Cordis, vědeckých zpráv z projektových stránek a souhrnných zpráv. Bylo zjištěno, že získání informací o projektu bylo zrychleno 7krát a zpracování zpráv 2krát oproti situaci, kdy je paralelnímu zpracování vypnuto.

Navržený systém tedy zprostředkovává o 136 % více vědeckých zpráv, než je dostupných na portálu Cordis. Tento fakt byl ovlivněn také tím, že bylo nalezeno o 182 % více webových stránek projektu, které algoritmus považuje za stránku projektu. Nicméně pouze na 10 % dohledaných stránek byly nalezeny vědecké zprávy.

Oproti prototypu je také značně rychlejší a umožňuje naplnit databázi za přijatelnou dobu.

4.3 Uživatelská stránka

Implementovaný portál nabízí ve srovnání jak s prototypem, tak s portálem Cordis na první pohled daleko propracovanější fasetové vyhledávání, které umožňuje i běžnému uživateli daleko rychleji nalézt požadované informace bez nutnosti ručně formulovat dotazy.

4.4 Přínos pro administrátora

Vzhledem k omezeným možnostem automatického přiřazování dat (zprávy a dohledaný odkaz na webovou stránku projektu), byl kladen obrovský důraz na možnost validace a oprav těchto dat. Webové prostředí umožňuje tyto případné problémy třídit, interpretovat a pohodlně opravit.

Kapitola 5

Závěr

Cílem této práce bylo především zlepšit kvalitu extrakce již existujícího systému a rozšířit jej tak, aby zjišťoval případné problémy a upozorňoval na ně administrátora. Stanovených cílů se mi podařilo dosáhnout a vytvořil jsem systém, který je nejen plně použitelný jako celek, ale i poskytuje administrátorovi kontrolu nad stavem celé databáze a umožňuje mu řešit problémy, které vycházejí od automatického přiřazování dat.

5.1 Zhodnocení práce

Největším přínosem mé práce je vytvoření systému, který je schopen získat a nabídnout více dat, než obsahuje oficiální webový portál o evropských projektech, čehož bylo dosaženo automatickými metodami. Systém si navíc umí poradit s velmi nespolehlivým a pomalým zdrojem dat a poskytuje v porovnání s portálem Cordis rychlejší a organizovanější přístup k informacím o evropských výzkumných projektech. Na druhou stranu poskytuje administrátorovi vhodné rozhraní a dostatek informací, na základě kterých může opravovat chybně přiřazené položky.

Z důvodu používaní pro mne nových technologií a neznámého portálu Cordis, bylo pro mne složité předvídat některé problémy, kvůli kterým jsem byl nucen některé části kódu přepsat, někdy i několikrát.

5.2 Další rozvoj práce

V následujících bodech se zaměřím na možná vylepšení a rozšíření, která by mohla vést k dalšímu zkvalitnění této práce.

- uložení dat v databázi bez redundance s využitím spojování dotazů za účelem ušetření místa
- využití dokonalejšího extraktoru vědeckých zpráv z webových stránek, který je schopen nalézt vědecké zprávy s vyšší úspěšností
- implementování faset nyní založených na funkci `facet()` z knihovny `elasticutils` pomocí agregace, čímž dojde ke zrychlení výpočtu těch faset, kde se fasetové hodnoty skládají ze dvou položek (druhá je obvykle zkratka)
- rozšíření počtu podporovaných formátů vědeckých zpráv, ze kterých je extrahován prostý text, za zmínku stojí zejména formáty PNG a JPG

- vylepšení pravděpodobnosti přiřazení správně dohledané webové stránky projektu

Literatura

[1] Apple Computer, Inc. *Search Basic*. Mac Developer Library [online]. 2005-06-12 [cit. 2016-05-09].
Dostupné z:
https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/SearchKitConcepts/searchKit_basics/searchKit_basics.html

[2] BARTOŠEK, Miroslav. Nástroje Google. 2. Google Scholar. *Zpravodaj ÚVT MU*, ročník 19, č. 2, 2008, s. 12–16. ISSN 1212-0901.
Dostupné z: http://ics.muni.cz/bulletin/clanky_tisk/602.pdf

[3] CLARKE, Irvine. Web-based B2B portals. *Industrial Marketing Management*, ročník 32, č. 1, 2013, s. 15–23. ISSN 0019-8501, doi:10.1016/S0019-8501(01)00199-7.
Dostupné z:
http://www.sciencedirect.com/science/article/pii/S0019850101001997

[4] DVOŘÁKOVÁ, Lucie. *Automaticky aktualizovaný webový portál o evropských výzkumných projektech*. Brno, 2015. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce: Smrž Pavel.
Dostupné z: https://www.vutbr.cz/studium/zaverecne-prace?zp_id=88606

[5] Elastic. *Inverted Index*. Elastic – Revealing Insights from Data [online]. ©2016 [cit. 2016-05-09].
Dostupné z: https://www.elastic.co/guide/en/elasticsearch/guide/current/inverted-index.html

[6] Elastic. *Query and filter context*. Elastic – Revealing Insights from Data [online]. ©2016 [cit. 2016-05-09].
Dostupné z: https://www.elastic.co/guide/en/elasticsearch/reference/2.1/query-filter-context.html

[7] Elastic. *You Know, for Search...*. Elastic – Revealing Insights from Data [online]. ©2016 [cit. 2016-05-09].
Dostupné z:
https://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html

[8] EMC Corporation. *Objem dat na světě se každé dva roky více než zdvojnásobí – vznikají tak nové příležitosti v oblasti Big Data a nové role v IT*. EMC Czech Republic [online]. 2011 [cit. 2016-05-11].
Dostupné z: http://czech.emc.com/about/news/press/2011/20110628-01.htm

[9] HELLER, Stanislav. *Knihovna pro podporu vývoje systému ReReSearch*. Brno, 2011. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce: Smrž Pavel.

[10] L-production. *Užitečné rady pro psaní odborného textu*. L-production [online]. ©2016 [cit. 2016-05-07].
Dostupné z: http://www.lproduction.cz/internetove-portaly-93.htm

[11] MITCHEL, Ryan. *Web Scraping with Python: Collecting Data from the Modern Web*. USA: O'Reilly Media, 2015. ISBN 9781491910276.
Dostupné z: https://books.google.cz/books?id=OUDfCQAAQBAJ

[12] SIRIISURIY, Mihiri. *A Comparative Study on Web Scraping*. International Research Conference [online]. 2015-11 [cit. 2016-05-07].
Dostupné z: http://www.kdu.ac.lk/proceedings/irc2015/2015/com-020.pdf

[13] SOBOTKA, Petr. *Nástroj pro optimalizaci fasetové navigace*. Praha, 2012. Diplomová práce. České vysoké učení technické v Praze. Vedoucí práce: Matyáš Petr.
Dostupné z:
http://www.diplomovaprace.cz/2012/13/DP_Sobotka_Petr_2012.pdf

[14] VLČEK, Lukáš. *Elasticsearch: Vyhledáváme hezky česky*. Zdroják – o tvorbě webových stránek a aplikací [online]. 2013-07-01 [cit. 2016-05-09].
Dostupné z:
https://www.zdrojak.cz/clanky/elasticsearch-vyhledavame-cesky/

[15] WINKLER, Ramona. *What is a Web Portal?* Atlantic Web Fitters [online]. ©2016 [cit. 2016-05-04].
Dostupné z: http://www.atlanticwebfitters.ca/AboutCMS/WhatisaWebPortal/tabid/95/Default.aspx

Přílohy

Seznam příloh

Příloha A

Obsah CD

- Zdrojové kódy implementovaného systému v adresáři `/src/`
- Tato práce ve formátu PDF v adresáři `/thesis/`
- Zdrojové kódy této práce ve formátu systému LaTeXv adresáři `/thesis/latex/`
- Plakát soubor `poster.pdf`